With OpenMP

Category: Porting to Pleiades

DRAFT

This article is being reviewed for completeness and technical accuracy.

Building Applications

To build an OpenMP application, you need to use the *-openmp* Intel compiler flag:

```
%module load comp-intel/11.1.072
%ifort -o your_executable -openmp program.f
```

Running Applications

The maximum number of OpenMP threads an application can use on a Pleiades node depends on (i) the number of physical processor cores in the node and (ii) if hyperthreading is available and enabled. Hyperthreading technology is not available for the Harpertown processor type. It is available and enabled at NAS for the Nehalem-EP and Westmere-EP processor types. With hyperthreading, the OS views each physical core as two logical processors and can assign two threads to it. This is beneficial only when one thread does not keep the functional units in the core busy all the time and can share the resources in the core with another thread. Running in this mode may take less than 2 times the walltime compared to running only 1 thread on the core.

```
Before running with hyperthreading for your production runs, it is
recommended that you experiment with it to see if it is beneficial
for your application.
```

Maximum Threads

Processor Type	Maximum Threads without Hyperthreading	Maximum Threads with Hyperthreading
Harpertown	8	N/A
Nehalem-EP	8	16
Westmere-EP	12	24

Here is sample PBS script for running OpenMP applications on a Pleiades Nehalem-EP node without hyperthreading:

```
#PBS -lselect=1:ncpus=8:ompthreads=8:model=neh,walltime=1:00:00
```

```
module load comp-intel/11.1.072

cd $PBS_O_WORKDIR

./your_executable
```

Here is sample PBS script with hyperthreading:

```
#PBS –lselect=1:ncpus=8:ompthreads=16:model=neh,walltime=1:00:00

module load comp-intel/11.1.072

cd $PBS_O_WORKDIR

./your_executable
```